

UNIVERSITÀ DI PISA



Facoltà di Ingegneria

Laurea Specialistica in Ingegneria dell'Automazione

Tesi di laurea

**Progettazione ed implementazione di un sistema
di obstacle avoidance per veicoli terrestri
autonomi basato su visione stereoscopica**

Candidato:

Alessandro Tortelli

Relatori:

Ch.mo Prof. Ing. Lorenzo Pollini

Ch.mo Prof. Ing. Mario Innocenti

Ing. Roberto Mati

Sessione di Laurea del 19/06/2006

Anno accademico 2005/2006

Sommario

L'argomento trattato nel presente lavoro è il riconoscimento ostacoli da parte di un veicolo autonomo che si muove in ambiente esterno, mediante visione artificiale. Per rilevare la distanza degli ostacoli è stata utilizzata la visione stereoscopica. L'approccio utilizzato può essere spiegato nelle seguenti fasi: 1) Estrazione di features dalle immagini provenienti dal sistema stereo 2) Individuazione delle corrispondenze tra le features provenienti da coppie di immagini stereo. 3) Triangolazione delle features di cui si è determinata una corrispondenza in modo da ottenere un insieme di punti nello spazio 3D 4) Clusterizzazione dei punti tridimensionali ottenuti dopo la triangolazione. Ogni cluster ottenuto rappresenta un'area non navigabile. 5) Creazione di una rappresentazione bidimensionale vista dall'alto degli ostacoli 6) Creazione della mappa bidimensionale vista dall'alto degli ostacoli. La mappa è stata creata sia relativamente al sistema di riferimento mobile solidale al veicolo, basandosi solo sullo stato corrente dell'ambiente che si presenta davanti al sistema di visione, sia relativamente ad un sistema di riferimento fisso, basandosi sui dati memorizzati durante tutto il tragitto che il veicolo ha percorso.

Abstract

In this paper, we address the problem of obstacles recognition for a vehicle that moves in an external environment, using computer vision. A stereoscopic vision system is used to estimate the distance of the obstacles. The proposed solution can be explained in the following steps: 1) extraction of features from the stereo frames. 2) Matching of features extracted in the previous step 3) Triangulation of the features matched in the previous step. 4) Clustering of 3D points computed in the previous step. Every cluster is a non-navigable area. 5) Creation of a 2D bird's eye representation of obstacles. 6) Creation of a 2D bird's eye map of the obstacles. The map was created relative to mobile reference positioned on the vehicle, using the information of the state of the environment in front of the vision system, and relative to fixed reference, using the information computed during all path.

| | |
|---|----|
| Sommario..... | 2 |
| Abstract..... | 2 |
| 1.INTRODUZIONE | 5 |
| 1.1 Premessa | 5 |
| 1.2 Scopo del lavoro | 5 |
| 1.3 Stato della conoscenza..... | 6 |
| 1.3.1 Utilizzo di sensori di visione | 6 |
| 1.3.2 Utilizzo di laser per il riconoscimento oggetti e creazione mappe..... | 9 |
| 1.3.3 Clustering..... | 16 |
| 2. LA SOLUZIONE PROPOSTA | 22 |
| 2.1 Approccio al problema dell'obstacle avoidance..... | 22 |
| 2.2 L' hardware utilizzato..... | 24 |
| 2.2.1 Il veicolo autonomo: Kart Ulisse..... | 24 |
| 2.2.2 Sensori di visione..... | 25 |
| 2.3 Ricostruzione delle coordinate nello spazio 3D | 38 |
| 2.4 Problema delle corrispondenze..... | 40 |
| 2.5 Fasi del progetto | 40 |
| 3. L' ALGORITMO SIFT (SCALE INVARIANT FEATURE TRANSFORM) | 43 |
| 3.1 Premessa | 43 |
| 3.2 Il SIFT (Scale Invariant Feature Transform) | 43 |
| 3.2.1 Rilevamento degli estremi scala-spaziali (Scale-space estrema detection)..... | 44 |
| 3.2.2 Localizzazione dei punti chiave..... | 46 |
| 3.2.3 Assegnamento dell'orientazione..... | 47 |
| 3.2.4 Descrittori dei punti chiave (Keypoint Descriptor) | 48 |
| 3.3 Matching tra features estratte con il SIFT | 49 |
| 3.4 Studio del funzionamento dell' algoritmo SIFT mediante esperimenti..... | 51 |
| 3.4.1 Esperimenti | 51 |
| 3.4.2 Tempo d'esecuzione | 62 |
| 3.5 Soluzioni ai problemi presentati dal SIFT | 62 |
| 3.5.1 Segmentazione | 62 |
| 3.5.2 Template Matching | 65 |
| 4. CREAZIONE DI UN DATABASE DI FEATURES..... | 71 |
| 4.1 Premessa | 71 |
| 4.2 Proprietà del descrittore SIFT..... | 72 |
| 4.2.1 Il descrittore di features ideale..... | 72 |
| 4.2.2 Il descrittore reale | 73 |
| 4.3 Implementazione..... | 75 |
| 4.3.1 Strumenti utilizzati..... | 75 |
| 4.3.2 Programma di gestione del database..... | 75 |
| 4.3.3 Strutture dati | 77 |
| 4.3.3 Aggiunta degli elementi al database | 78 |
| 4.4 Risultati..... | 79 |
| 5. CLUSTERIZZAZIONE FEATURES E CREAZIONE MAPPA | 81 |
| 5.1 Premessa | 81 |
| 5.2 Clusterizzazione delle features | 81 |
| 5.2.1 Scelta del metodo di clusterizzazione..... | 81 |
| 5.2.2 Il metodo utilizzato | 82 |

| | |
|--|-----|
| 5.2.3 Implementazione..... | 83 |
| 5.2.4 Scelta del raggio delle sfere | 84 |
| 5.3 Rappresentazione degli ostacoli | 89 |
| 5.3.1 Il problema della rappresentazione bidimensionale | 89 |
| 5.3.2 Il metodo di rappresentazione proposto..... | 89 |
| 5.3.3 Semplificazione della rappresentazione | 95 |
| 5.4 Costruzione della mappa degli ostacoli | 99 |
| 5.4.1 Posizione delle features nel sistema di riferimento fisso..... | 99 |
| 5.4.2 Selezione delle features mediante clusterizzazione..... | 101 |
| 5.4.3 Tracking dei clusters nel tempo | 103 |
| 5.4.4 Selezione delle features mediante finestra temporale..... | 104 |
| 5.4.5 Risultati..... | 106 |
| 5.5 Chiusura dell'anello..... | 108 |
| 5.5.1 La struttura dati di uscita | 108 |
| 5.5.2 Implementazione..... | 108 |
| 5.5.3 Risultati..... | 109 |
| 6. CONCLUSIONI | 111 |
| BIBLIOGRAFIA | 112 |

1.INTRODUZIONE

1.1 Premessa

I veicoli autonomi, o più in generale i robot, operano oramai su una larga scala di settori. Ciononostante il loro utilizzo risulta pressoché assente nella maggior parte delle applicazioni in cui non si ha una conoscenza completa o certa dell'ambiente di lavoro.

Gli studi compiuti negli ultimi due decenni offrono oggi in commercio una larga varietà di sensori in grado di determinare le caratteristiche dell'ambiente di lavoro dei robot. In particolare nella Ricerca assume un ruolo fondamentale lo studio e il perfezionamento di *sensori di visione* per questo scopo, dal momento che questi consentono di poter *misurare* l'ambiente senza nessun tipo di contatto.

I *sistemi di visione* tentano di riprodurre il senso della vista dell'uomo, basandosi sulle immagini provenienti da una o più telecamere. La sempre crescente velocità degli elaboratori e il perfezionamento delle tecniche di analisi delle immagini consentono elevate prestazioni anche in caso di telecamere a costo ridotto.

In generale la visione e l'azionamento che deriva dai sottosistemi atti a questi scopi si combinano insieme secondo un modello *open-loop* del tipo *looking-then-moving*. L'accuratezza del risultato dell'operazione dipende strettamente dalla precisione dei sistemi di visione (nel caso dei robot anche dall'accuratezza dell'*end-effector*). Per incrementarne l'accuratezza è possibile utilizzare un approccio cosiddetto *visual-servoing*, basato sul modello del tipo *visual-feedback control-loop*, in questo caso si fondono insieme le conoscenze provenienti da diversi campi quali l'elaborazione delle immagini ad alta velocità, la cinematica, la dinamica, la teoria del controllo, e l'elaborazione real time.

L'utilizzo esclusivo di un unico sensore di Visione (per esempio una sola telecamera) nel calcolare le distanze degli oggetti o degli ostacoli che si trovano nell'ambiente di lavoro del veicolo è però problematico. Da una sola immagine non è possibile ricavare informazioni sulla profondità, se non è nota a priori almeno una delle dimensioni di uno degli oggetti visualizzati.

Per ovviare al problema, esistono sistemi che utilizzano, per valutare le distanze, la tecnica della Visione Stereoscopica, tipica dell'uomo, implementata ad esempio sugli ultimi veicoli inviati dalla NASA su Marte.

Spesso sui veicoli esiste un'integrazione tra visione e odometria. Confrontando le distanze percorse e le relative modifiche subite dagli oggetti nell'immagine acquisita può essere valutata la mappa dell'ambiente di lavoro del veicolo.

Un altro sensore di largo uso nella Ricerca che permette di misurare l'ambiente senza contatto è il *laser*. Esso permette di conoscere la profondità degli oggetti che si trovano intorno al robot ma è un sensore attivo dal costo più elevato rispetto al sensore di visione. Inoltre il laser non permette di riconoscere gli oggetti in base al loro aspetto. La possibilità di riconoscere gli oggetti in base al loro aspetto risulta utile nel caso ci sia la necessità di rilevare il movimento degli oggetti.

1.2 Scopo del lavoro

In questo lavoro sono state prese in analisi le tecniche e le varie problematiche che concernono il riconoscimento degli ostacoli da parte di un veicolo autonomo in ambiente esterno non strutturato mediante l'utilizzo di visione stereoscopica, al fine di creare una mappa degli

ostacoli che si trovano nell'ambiente dinamico in cui il robot naviga. Per ambiente dinamico si intende un ambiente in cui non solo il robot ma anche altri corpi si muovono.

1.3 Stato della conoscenza

In questo paragrafo sarà illustrato lo stato dell'arte sia per quanto riguarda le varie tecniche implementate nel presente lavoro sia per quanto riguarda il problema del riconoscimento ostacoli e mapping di ambienti non strutturati. Le tecniche più utilizzate nella Ricerca per risolvere il problema del riconoscimento ostacoli e creazione mappa in ambiente non strutturato sono quelle dell'utilizzo dei sensori di visione e dell'utilizzo di laser. Tali tecniche saranno analizzate di seguito.

1.3.1 Utilizzo di sensori di visione

1.3.1.1 La visione nel campo dell'automazione

Gli scopi di questo lavoro di tesi portano a focalizzare l'attenzione sulla Computer Vision nel settore dell'Automazione per Veicoli Autonomi (*Autonomous Vehicle*, AV).

Gli AV vengono oramai largamente utilizzati sia per eseguire dei processi di routine per l'industria, sia per svolgere compiti particolarmente azzardati per l'uomo e non solo. La *Machine Vision* (Visione attraverso le macchine) tenta di "donar la vista" ai veicoli, al fine di *capire* il loro ambiente circostante, consentendo un uso più flessibile degli stessi.

Nel 2003 AIA (Automated Image Association) ha stimato il volume d'affari dell'intera industria della Computer Vision aggirarsi intorno ai 6.6 bilioni di dollari (con un aumento del 14% rispetto al 2002). Una grossa porzione di questa somma si riferisce poi agli investimenti specifici nel campo dell'Automazione, come è possibile rilevare, ad esempio dai dati forniti da David Lowe nel suo sito.

Tuttavia, anche se la velocità e la potenza dei calcolatori è in continua crescita, la Ricerca è ancora lontana dal riuscire a processare le immagini come è in grado di fare il genere umano. Il cervello umano è in grado di analizzare gli elementi 10 volte più lentamente che i chip digitali, eppure in tempo reale, grazie a una qualche elaborazione parallela e grazie ad un largo uso di astrazioni gerarchiche (anche se tali fenomeni ancora non sono chiari).

Cercando di schematizzare la visione umana, questa consiste di due sensori di visione (gli occhi), i quali lavorano in stretto legame con un interprete, il cervello. Entrambi devono essere replicati per dotare le macchine di una capacità visiva. Fortunatamente la maggiorparte delle applicazioni industriali non richiedono di coprire il vasto panorama della sensitività della visione umana, ad esempio, spesso è sufficiente riconoscere solo la sagoma di un oggetto.

I sensori maggiormente utilizzati sono le videocamere, in un numero variabile a seconda degli scopi. In alcuni casi queste vengono affiancate da sensori come scanner laser, ma in questo ambito ci limitiamo all'uso esclusivo di videocamere.

I tre casi applicativi su cui si riversa la maggior parte della Ricerca in campo Visuale-Automatico trattano :

Rilevamento degli ostacoli (*Obstacle Avoidance*): La Visione stereo (tramite cioè 2 videocamere) consente agli AV di capire il loro ambiente di lavoro in 3 dimensioni. In questo modo è possibile ricostruire una mappa dell'ambiente libero da ostruzioni (*Free Space Mapping*) o, in maniera complementare, una mappa degli ostacoli rilevati (*3D Depth Map*), in modo che il veicolo autonomo sia in grado di individuare una chiara traiettoria tra questi. I contorni dello scenario 3D, derivati attraverso la stereovisione, possono essere usati per la navigazione del veicolo e per la localizzazione e tracciamento degli oggetti conosciuti.

Esplorazione dell'ambiente sconosciuto circostante (*Exploring New Surroundings*): Se un veicolo guidato autonomo (AGV) si muove in un ambiente sconosciuto, deve essere in grado di riconoscerne le caratteristiche, intanto che procede. Un approccio prevede l'analisi delle immagini provenienti da una videocamera posizionate sulla parte anteriore del veicolo. Innanzitutto vengono estratte le feature come gli angoli (giunzioni a T e giunzioni a Y), che sono localizzate come l'AGV si muove sulla scena. Tali feature vengono scelte, perché il punto di intersezione delle linee che formano l'angolo viene fissato rispetto all'angolo da cui viene visto. Le feature vengono poi tracciate nella serie di immagini. Come l'AGV si muove, il moto apparente delle feature in questo campo di vista dipenderà dalla distanza. Infatti dalla traiettoria delle feature dell'immagine, la loro posizione 3D e il moto 3D dell'AGV possono venir stimati. Sovrapponendo una griglia sull'immagine può essere definita una regione in cui guidare il veicolo.

Sorveglianza di ambienti conosciuti (*Surveillance*): I veicoli utilizzati sono dotati di un meccanismo per localizzare la loro posizione relativa all'ambiente circostante. Il calcolo esatto della posizione dell'AV è essenziale per la performance sicura e efficace dei task. E' importante monitorare i movimenti dell' AV e mettere in relazione queste informazioni con la stima che l'AV possiede della sua posizione. Alcuni sistemi usano 4 videocamere per rilevare l'ambiente di lavoro. L'immagine di ciascuna videocamera viene usata per identificare gli oggetti/ostacoli sottraendo l'immagine ricevuta rispetto ad una immagine di riferimento che raffigura l'ambiente di lavoro "vuoto". I dati dalle quattro videocamere vengono fuse per ottenere una locazione precisa. Gli oggetti, ad esempio le persone, possono essere riconosciute da un AV, facendo uso di modelli che descrivano le feature caratteristiche.

In questo lavoro di tesi prendiamo in considerazione soprattutto il primo caso (Obstacle avoidance), trattandosi appunto dello scopo principale del lavoro svolto.

1.3.1.2 Vantaggi

La scelta di un sensore di tipo videocamera per la realizzazione di un Sistema di Visione è motivata dai parecchi vantaggi che ne derivano; i principali risultano:

- Ricchezza di informazioni: Grazie ai sensori di visione, è possibile ottenere informazioni sull'aspetto degli oggetti; cosa che non è possibile avere se si utilizza altri sensori come i laser (par.1.3.2).
- Sensore passivo (*passive sensor*): Non è possibile rilevare la presenza del sensore dal momento che l'ambiente viene misurato senza nessun tipo di contatto ed è necessaria poca potenza; diversamente da ciò che accade qualora si scelgano sensori attivi, ad esempio il laser (par. 1.3.2) o il radar.
- Dimensioni: Esistono videocamere in commercio anche molto piccole, il che non dà problemi anche in caso di miniAV o microAV (veicoli autonomi di ridotte dimensioni).

- Costo: Il costo di una videocamera è divenuto oramai molto contenuto, neanche lontanamente paragonabile rispetto alla maggior parte degli altri sensori, come laser o radar. Per tante applicazioni di visione non è necessaria neanche un'alta risoluzione ed è possibile far uso di webcam.
- Facilità di interpretazione per un utente: In molte applicazioni anche una sola videocamera fornisce informazioni sufficienti e persino intuitive, nel caso sia necessario un'interfaccia con un utente umano.

1.3.1.3 Visione per il riconoscimento oggetti e creazione di mappe

Per dare un'idea dello stato dell'arte nel campo della visione per il riconoscimento ostacoli e creazione mappa esaminiamo alcuni articoli che trattano l'argomento.

Nell'articolo [7] datato 2002 viene affrontato il problema del riconoscimento ostacoli mediante visione stereoscopica in ambienti interni. Una cosa importante da notare è che il sistema funziona solo con telecamere rivolte verso il basso. In questo modo ci si porta in una situazione semplificata perché il campo visivo è più ristretto (Fig.1.1) rispetto ad un sistema in cui le telecamere sono rivolte verso l'orizzonte come quello proposto nella presente tesi.



Fig.1.1 Robot utilizzato nell'articolo [7].Notare che le telecamere sono rivolte verso il basso.

La soluzione proposta nell'articolo consiste nei seguenti passi:

1. Individuare il pavimento sulla coppia di immagini
2. Eliminare il pavimento dalle immagini e segmentare le immagini rimaste in modo da ottenere gli ostacoli.
3. Proiettare gli ostacoli sul pavimento in modo da ottenere una mappa che consenta di effettuare una pianificazione del percorso.

Essendo il campo visivo limitato anche la mappa degli ostacoli sarà di grandezza limitata. Il tempo d'esecuzione è buono (pari a circa 0.1 sec per l'intero ciclo di riconoscimento ostacoli e creazione mappa) ma i test sono stati effettuati solo in ambienti interni.

Nell'articolo [8], datato 2005, viene affrontato il problema della localizzazione del robot e della creazione della mappa di un ambiente interno statico. Il problema viene risolto mediante visione stereoscopica utilizzando tre telecamere. Viene utilizzato l'algoritmo SIFT per estrarre features dalle immagini. L'algoritmo SIFT verrà spiegato nel dettaglio nel capitolo 3. Dopo aver estratto features dai tre frames che, nello stesso istante, provengono dalle tre telecamere, si

trovano le corrispondenze tra le features e si triangolano le features corrispondenti in modo da ottenere un insieme di punti nello spazio tridimensionale. Il sistema di stereovisione utilizza tre telecamere, invece delle due utilizzate solitamente, per filtrare meglio le eventuali corrispondenze tra features errate. La mappa creata non è altro che un insieme di punti bidimensionali che non vengono clusterizzati.

Esiste la possibilità di trovare una corrispondenza tra features estratte dal SIFT all'interno di un database grazie ad un descrittore di feature che l'algoritmo SIFT calcola. (vedi capitolo 4) Sfruttando tale descrittore viene creato un database di features ed utilizzato per la localizzazione del robot.

Nell'articolo [9], datato 2005, si affronta il problema del tracking di oggetti su sequenze video. La soluzione proposta consiste nell'estrarre features dalle immagini che compongono il video e creare un database di queste features. Il database ci permette di eseguire il tracking delle features sulla sequenza video. Per l'estrazione delle features viene usato l'algoritmo SIFT; il tracking viene realizzato sfruttando i descrittori restituiti dal SIFT. Le features vengono clusterizzate: un insieme di features rappresenta un oggetto; seguire un insieme di features equivale a seguire un oggetto.

Il sistema, monoculare, funziona solo se il sensore di visione è lontano dagli oggetti da seguire.

1.3.2 Utilizzo di laser per il riconoscimento oggetti e creazione mappe

I laser sono largamente usati per il riconoscimento ostacoli soprattutto in ambienti dinamici data l'elevata precisione ed il campo visivo molto più ampio di quello che si ha utilizzando sensori di visione.

Per quanto riguarda la presente argomentazione, ci basta sapere che il laser è un dispositivo che ci permette di fare una scansione dell'ambiente circostante per misurare la profondità cui gli oggetti si trovano. In questo modo, è possibile ottenere una vista dall'alto (o bird's eye view) dell'ambiente circostante che ci permette di valutare la distanza degli ostacoli (vedi fig.1.2)

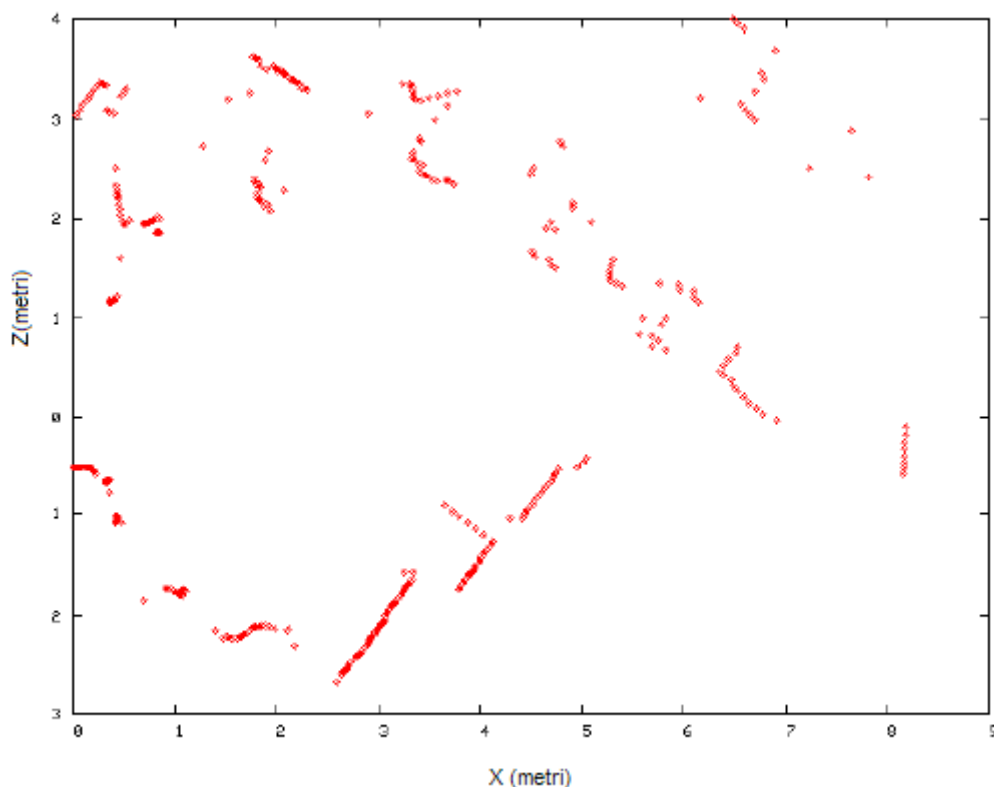


Fig.1.2 Esempio di dati provenienti da una scansione laser con lo scanner in posizione (0,0)

In fig.1.2 sono mostrati i dati, riguardanti l'ambiente circostante, ottenuti da una scansione del laser.

Passiamo ad analizzare alcune recenti pubblicazioni sull'argomento.

Nell'articolo [1] datato 2005 viene preso in considerazione il problema dell'obstacle avoidance in ambienti dinamici. Per il riconoscimento degli ostacoli viene utilizzato un laser.

L'ambiente viene misurato mediante una serie di scansioni successive che ci danno una stima della sua struttura (Fig.1.2).

La soluzione proposta per risolvere il problema dell'individuazione degli ostacoli, consiste nel creare una griglia che rappresenta l'ambiente che sta intorno al robot. La griglia è realizzata trasformando le coordinate polari ottenute dal laser in coordinate cartesiane e rappresentando l'ambiente circostante di 30x30 metri mediante celle da 20x20 cm. Ogni cella può assumere tre stati:

1. occupata
2. libera
3. stato sconosciuto

E' stato utilizzato un algoritmo di segmentazione di immagini per clusterizzare le celle occupate e ottenere una lista degli ostacoli presenti. Notare che non viene specificato quale algoritmo di segmentazione è stato impiegato.

Una volta eseguita la clusterizzazione si ottiene una lista di ostacoli che è il risultato della scansione al tempo t .

Per realizzare un tracking degli ostacoli presenti tra una scansione e l'altra ci si basa sulla posizione del baricentro dei cluster.

Per predire la posizione del cluster nella scansione successiva viene fatta una stima della velocità mediante rapporto incrementale:

$$v_{cent_i} = (r_{cent_i}(t) - r_{cent_i}(t - 1)) / \Delta t \quad (1.1)$$

dove:

$r_{cent_i}(t)$ è il baricentro del cluster i al tempo t

v_{cent_i} è la velocità stimata del baricentro del cluster

La posizione del baricentro viene predetta mediante la formula:

$$r_{PREDICTED_{cent_i}}(t) = r_{cent_i}(t - 1) + v_{cent_i} \cdot \Delta t \quad (1.2)$$

dove $r_{PREDICTED_{cent_i}}$ è la posizione del baricentro predetta.

La posizione predetta dei baricentri dei clusters viene utilizzata per trovare una corrispondenza fra clusters ottenuti in due istanti successivi. Ad ogni cluster corrisponde un ostacolo.

Il diagramma di flusso di fig.1.3 esplica il funzionamento della tecnica qui esposta.

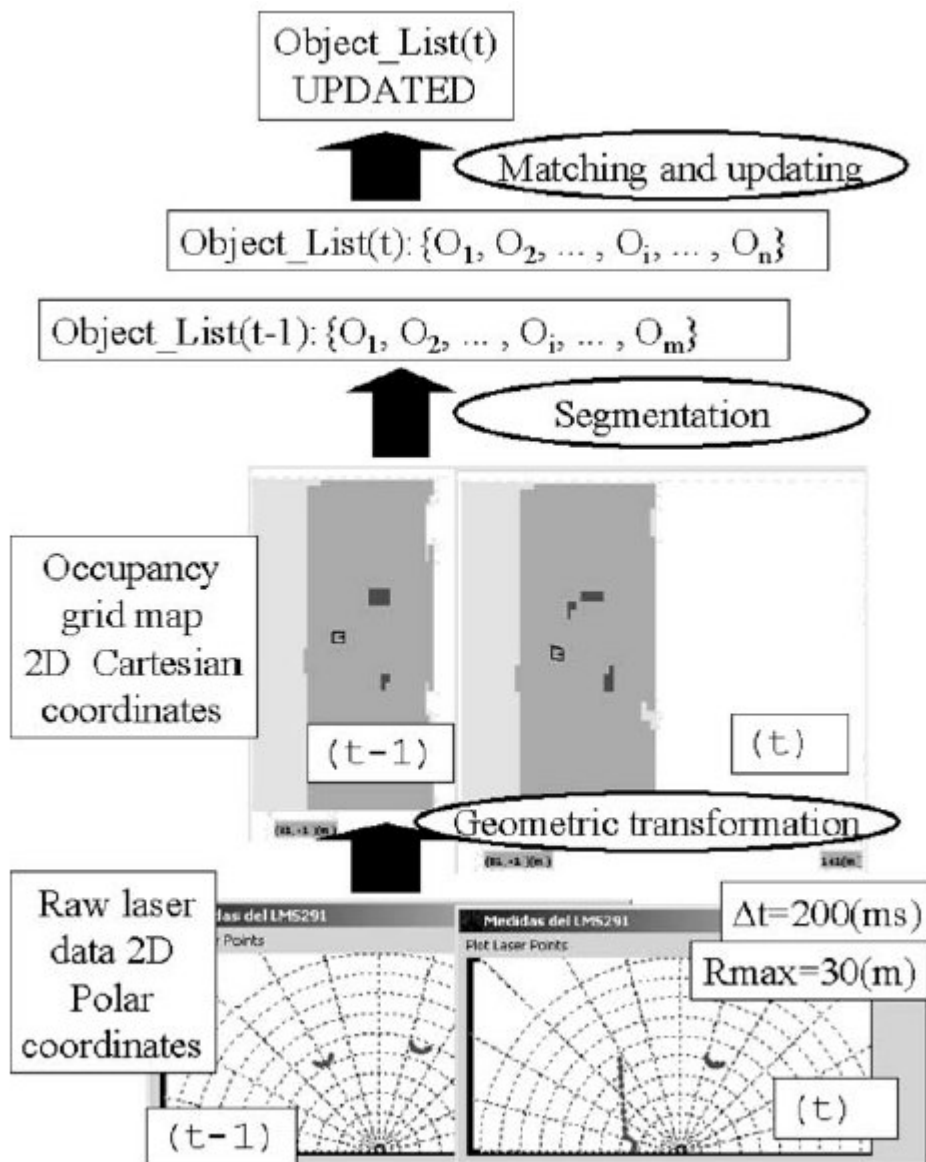


Fig. 1.3 Diagramma di flusso che esplica il funzionamento della tecnica di rappresentazione dell'ambiente sviluppata in [1]

Una volta calcolata la lista degli ostacoli viene attuato un controllo del veicolo al fine di realizzare l'obstacle avoidance. Il controllo viene realizzato mediante semplici regole fuzzy.

Secondo quanto scritto nell'articolo, il sistema realizzato in [1] riesce quindi ad eseguire il tracking degli ostacoli in maniera efficiente e con bassi costi dal punto di vista computazionale.

Le prove sperimentali sono state eseguite in ambienti esterni.

Nell'articolo [2] viene analizzato il problema della creazione di una mappa bidimensionale vista dall'alto degli ostacoli. Il problema viene affrontato mediante l'utilizzo di laser; il robot effettua scansioni ad intervalli regolari per determinare la struttura dell'ambiente. L'ambiente in cui il sistema progettato promette di funzionare è "quasi-statico" ossia un ambiente in cui gli oggetti si muovono solo occasionalmente e con dinamica costante come specificato nel sommario dell'articolo in questione.

Nell'introduzione vengono messe in evidenza le difficoltà che si incontrano nella creazione di mappe di ambienti sia statici sia dinamici; si sostiene che, in ambiente non strutturato, gli oggetti sono troppo vari per essere rappresentati con un numero fisso di primitive geometriche.

Viene affrontato il problema della clusterizzazione dei punti bidimensionali ottenuti dal laser (fig.1.4).

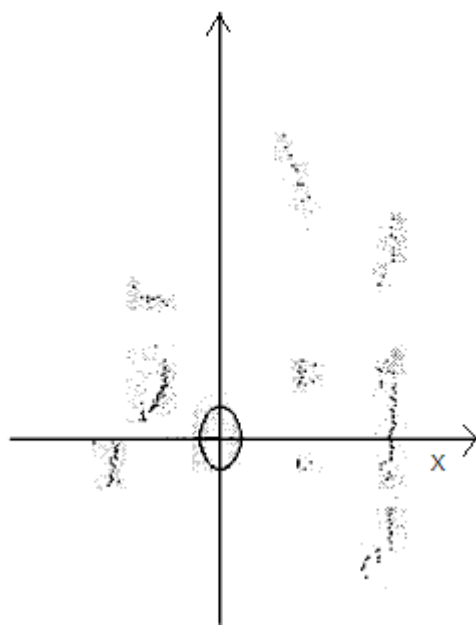


Fig.1.4 Dati da clusterizzare ottenuti tramite scansione laser. I dati sono punti bidimensionali che rappresentano l'ambiente che sta attorno al robot.

I dati vengono clusterizzati utilizzando la tecnica incrementale: due punti appartengono allo stesso cluster se la loro distanza euclidea sta sotto ad una certa soglia. Una cosa importante da notare è che la soglia è stata scelta in maniera euristica tenendo presente la grandezza del robot, come scritto nel paragrafo 2.1 dell'articolo.

Una volta che i dati sono stati clusterizzati, vengono calcolati i seguenti parametri per ogni cluster:

- baricentro del cluster
- matrice di covarianza dei punti bidimensionali che compongono il cluster
- autovettori della matrice di covarianza

Rappresentare i cluster con questi parametri equivale a rappresentarli come ellissi. (Fig.1.5)

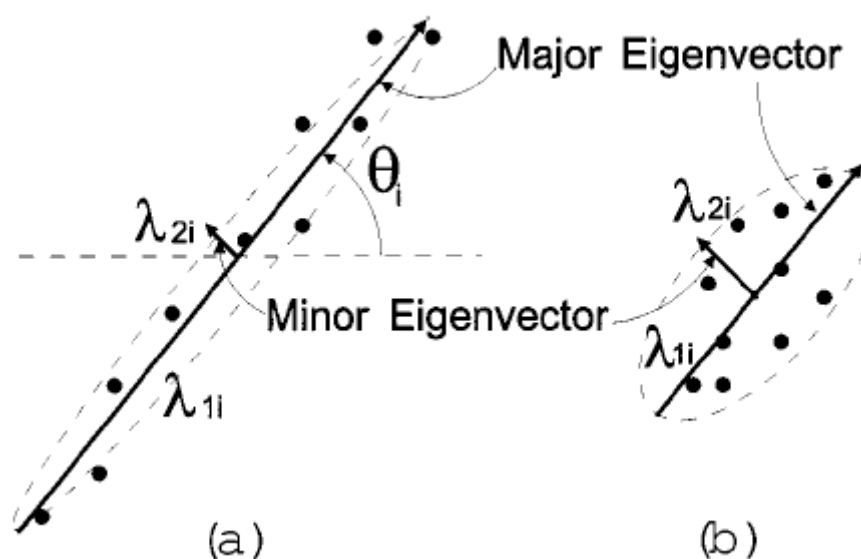


Fig.1.5 Cluster di punti che vengono rappresentati come ellissi

Questi dati vengono sfruttati sia per trovare una corrispondenza tra i cluster calcolati ad un istante ed i cluster calcolati all'istante precedente, sia per sommare, sottrarre o spostare i clusters dell'istante attuale con quelli dell'istante precedente. Per capire cosa significa "sommare sottrarre o spostare" basta vedere fig.1.6 che è esplicativa di come la mappa degli ostacoli venga aggiornata.

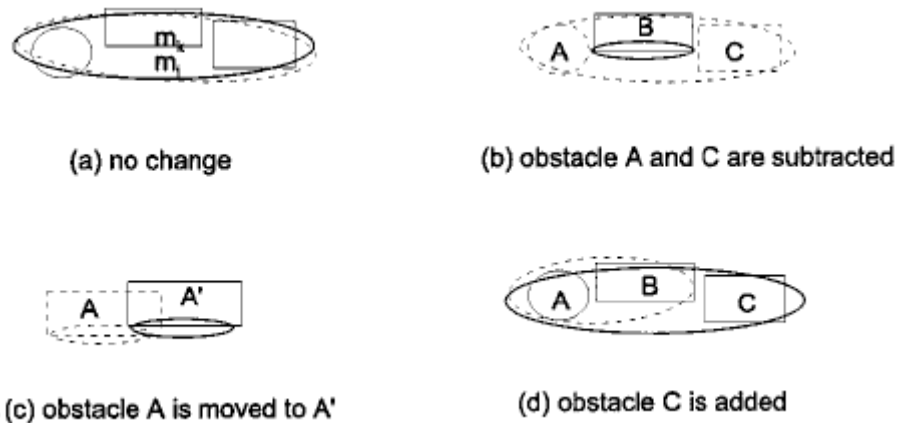


Fig.1.6 Schema esplicativo del metodo con cui viene aggiornata la mappa.

Linea tratteggiata: ostacoli e oggetti al passo precedente

Linea continua: ostacoli e oggetti al passo attuale

In fig.1.6 le ellissi sono la rappresentazione degli ostacoli mentre i quadrati e i cerchi sono gli oggetti effettivamente esistenti. Notare che più oggetti possono essere rappresentati come un unico ostacolo.

Gli esperimenti sono stati eseguiti in ambienti interni. Un ciclo di clustering dei dati e aggiornamento della mappa ha un tempo d'esecuzione pari a 200ms.

Nell'articolo [3] è affrontato il problema del tracking di oggetti in movimento nell'ambiente in cui si muove un robot. Per risolvere il problema è stato utilizzato un laser.

Per ogni scansione è stata creata una griglia come quella di fig. 1.6.

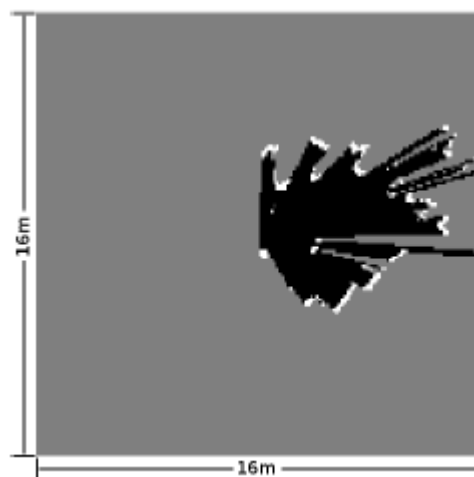


Fig.1.7 Scansione laser convertita in una griglia.

Il laser si trova al centro della griglia

Il movimento degli oggetti è stato rilevato in base alla differenza tra due scansioni successive. (vedi figura 1.8). Si ricorda che il laser utilizzato effettua una scansione di 180 gradi.

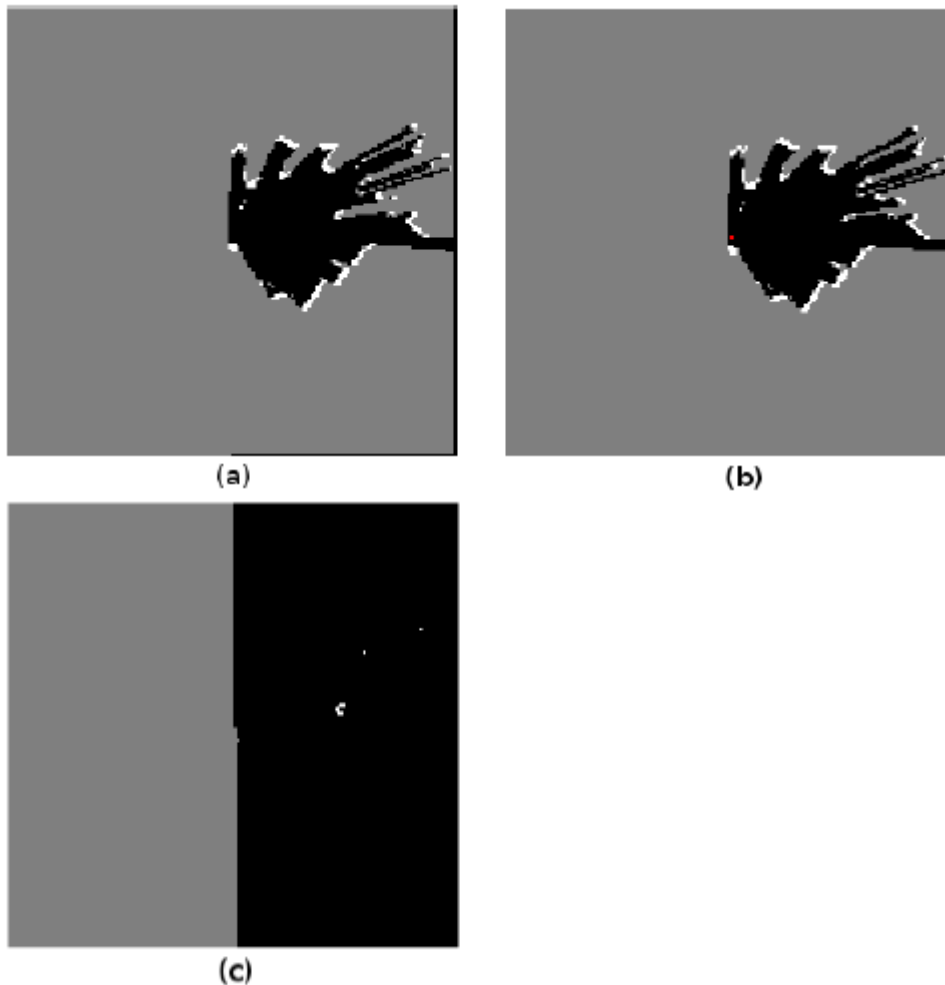


Fig.1.8 Due scansioni successive e differenza risultante. (a) Griglia corrente (b) Griglia precedente (c) Immagine che rappresenta la differenza delle due griglie. Le regioni in bianco rappresentano i cambiamenti. Le regioni in grigio rappresentano gli spazi non validi.

Le regioni in bianco nell'immagine risultato della differenza tra due scansioni (Fig.5c) vengono clusterizzate: i punti adiacenti appartengono allo stesso cluster, ogni cluster rappresenta un oggetto.

Per predire la posizione degli oggetti è stato usato un filtro di Kalman [Kalman, 1960].

Il modello utilizzato è il seguente:

$$x_t = x_{t-1} \quad (1.3)$$

dove:

x_t è la variabile di stato al tempo t

x_{t-1} è la variabile di stato al tempo $t-1$

La variabile di stato rappresenta la velocità dell'oggetto quindi per la (1.3) si ha la supposizione di velocità costante. Siccome tale supposizione non è valida in generale, sono state ipotizzate più velocità possibili per ogni oggetto.

I test del sistema progettato sono stati effettuati in ambienti interni.

1.3.3 Clustering

Il clustering è il problema di riunire un insieme di dati in gruppi. Il significato, di questi gruppi, dipende dall'applicazione.

Nella presente tesi il problema del clustering sarà affrontato quando ci sarà la necessità di raggruppare punti tridimensionali in modo che ogni gruppo (o cluster) formi un ostacolo.

La clusterizzazione è un problema che si incontra nei più svariati campi della tecnica ed i metodi per la sua soluzione sono svariati. Non c'è un metodo migliore per clusterizzare ma la scelta di un metodo rispetto ad un altro dipende soprattutto dall'applicazione.[10]

Gli algoritmi di clustering possono essere divisi in due grandi classi:

- algoritmi di clustering che prevedono la conoscenza a priori del numero di gruppi in cui i dati vanno riuniti.
- algoritmi di clustering che non prevedono la conoscenza a priori del numero di gruppi in cui i dati vanno riuniti.

Tra gli algoritmi in cui si prevede la conoscenza a priori del numero di cluster, possiamo annoverare i seguenti come i più largamente utilizzati e conosciuti [10]:

- K-Means
- Fuzzy C-Means
- Reti neurali competitive

Per quanto riguarda gli algoritmi in cui non c'è conoscenza a priori del numero di clusters elenchiamo i seguenti:

- Clustering Gerarchico
- Clustering Incrementale
- Metodi a potenziale

1.3.3.1 Specifiche

Le specifiche che in generale si richiedono ad un algoritmo di clustering sono le seguenti:

- L'algoritmo deve essere capace di riconoscere clusters di forma arbitraria
- L'algoritmo deve funzionare anche in presenza di rumore e di outliers
- L'algoritmo deve essere insensibile all'ordine con cui i dati di ingresso vengono forniti
- I risultati dell'algoritmo di clusterizzazione, che in molti casi possono essere arbitrari, devono essere facilmente interpretabili
- L'algoritmo deve essere scalabile
- L'algoritmo deve funzionare con dati di ingresso a molte dimensioni

1.3.3.2 Inconvenienti

Quando si ha a che fare con un algoritmo di clustering gli inconvenienti che solitamente si incontrano sono i seguenti:

- Le tecniche di clusterig esistenti non rispondono adeguatamente a tutte le specifiche. (par. 1.3.3.1)
- Lavorare con dati di grosse dimensioni può essere problematico a causa del costo computazionale.
- Per i clustering basati sulla distanza, l'efficacia del metodo dipende dalla definizione di distanza.
- Nel caso in cui una misura di distanza ovvia non esista, si deve definire una misura di distanza che non è facile se lavoriamo in spazi multidimensionali.
- Il risultato dell'algoritmo, che può essere arbitrario, può essere interpretato in diversi modi.

1.3.3.3 Metodo K-Means

K-Means [MacQueen,1967] è uno dei più semplici algoritmi che risolvono il problema del clustering. L'idea fondamentale consiste nel definire k centroidi, dove k è il numero di clusters scelto a priori.

Dato un insieme di vettori d'ingresso, ogni vettore è di dimensione n , tale insieme deve essere raggruppato in k clusters. Un centroide è un vettore di dimensione n che rappresenta il centro del cluster. L'algoritmo consiste nei seguenti passi:

1. Posizionare i k centroidi nello spazio a n dimensioni. Tale posizionamento rappresenta la scelta iniziale per i centri dei clusters.
2. Assegnare ad ogni vettore da clusterizzare al cluster che ha il centroide più vicino alla propria posizione.
3. Una volta che tutti i vettori sono stati assegnati ad un cluster, calcolare nuovamente la posizione dei k centroidi.
4. Ripetere i punti 2 e 3 finché i centroidi non si muovono più.

I principali svantaggi dell'algoritmo K-Means sono i seguenti:

- Non è specificato il modo in cui si possono inizializzare i centroidi. Un metodo popolare è quello di inizializzarli con valori casuali.
- Il risultato prodotto dipende dai valori d'inizializzazione scelti per i centroidi. Una soluzione standard di questo problema è provare con un certo numero di inizializzazioni differenti
- Il risultato dipende dalla scelta del numero di clusters k .

Lo svantaggio di maggior rilievo è l'ultimo visto che non c'è un metodo per trovare il numero ottimo di cluster, che valga in generale.

Una semplice soluzione del problema può essere quella di eseguire l'algoritmo con diversi valori di k e scegliere il miglior valore di k secondo certi criteri che definiscono la qualità della clusterizzazione. Bisogna stare molto attenti con questo approccio perché, aumentando il numero dei clusters k , la funzione di errore diminuisce per definizione ma aumenta il rischio di overfitting. Per fare un esempio, se il numero di elementi da clusterizzare è pari al numero di cluster, un risultato dell'algoritmo di clustering che manda la funzione di errore a zero può essere

quello in cui ogni elemento è centroide di un cluster. Tale soluzione manda la funzione di errore a zero, ma il risultato di questa clusterizzazione risulta inutilizzabile.

1.3.3.4 Metodo Fuzzy C-Means

Fuzzy C-Means è un metodo in cui un dato di ingresso può appartenere ad uno o più clusters. Viene definito un numero fuzzy chiamato grado di appartenenza che ci dice quanto un dato di ingresso appartenga ad un cluster.

Il funzionamento dell'algoritmo è simile al k-means ma solo che in questo caso si lavora in maniera fuzzy.

Rimane sempre il problema della scelta del numero di clusters.

1.3.3.5 Reti neurali competitive

Per risolvere il problema della clusterizzazione vengono spesso usate le reti neurali competitive [10]. Questo tipo di reti neurali ha un numero di neuroni pari al numero di clusters. Le reti neurali competitive vengono addestrate in modo che, al termine dell'addestramento, ogni neurone rappresenti il baricentro di un cluster.

Rimane sempre il problema della scelta del numero di clusters.

1.3.3.6 Clustering Gerarchico

Dati N elementi da clusterizzare, data la matrice $N \times N$ delle distanze tra gli elementi il processo di clustering gerarchico [S.C Johnson, 1967] può essere spiegato con i seguenti passi:

1. Inizialmente si assegna un cluster per ogni elemento da clusterizzare; perciò se si hanno N elementi da clusterizzare adesso avremo N clusters ciascuno composto da un elemento. La distanza tra i clusters equivale alla distanza tra gli elementi che contengono
2. Si trova la coppia di clusters più vicina. I due clusters che compongono la coppia vengono uniti in un unico cluster. Dopo questo passo si ha un numero di cluster ridotto di un unità.
3. Si calcola la distanza tra il nuovo cluster e tutti i vecchi clusters.
4. Si ripetono i punti 2 e 3 finché tutti gli elementi non sono riuniti in un unico cluster di dimensione N .

Vi sono tre possibili varianti di questo tipo di clustering che dipendono da come viene eseguito il passo 3:

- *Single-linkage*: la distanza tra due clusters è definita come la distanza minima che esiste tra gli elementi di un cluster e gli elementi dell'altro.
- *Complete-linkage*: la distanza tra due clusters è definita come la distanza massima che esiste tra gli elementi di un cluster e gli elementi dell'altro.
- *Average-linkage*: la distanza tra due clusters è definita come la distanza media che esiste tra gli elementi di un cluster e gli elementi dell'altro.

Il risultato dell'algoritmo di clustering gerarchico può essere schematizzato con un albero come in fig.1.9.

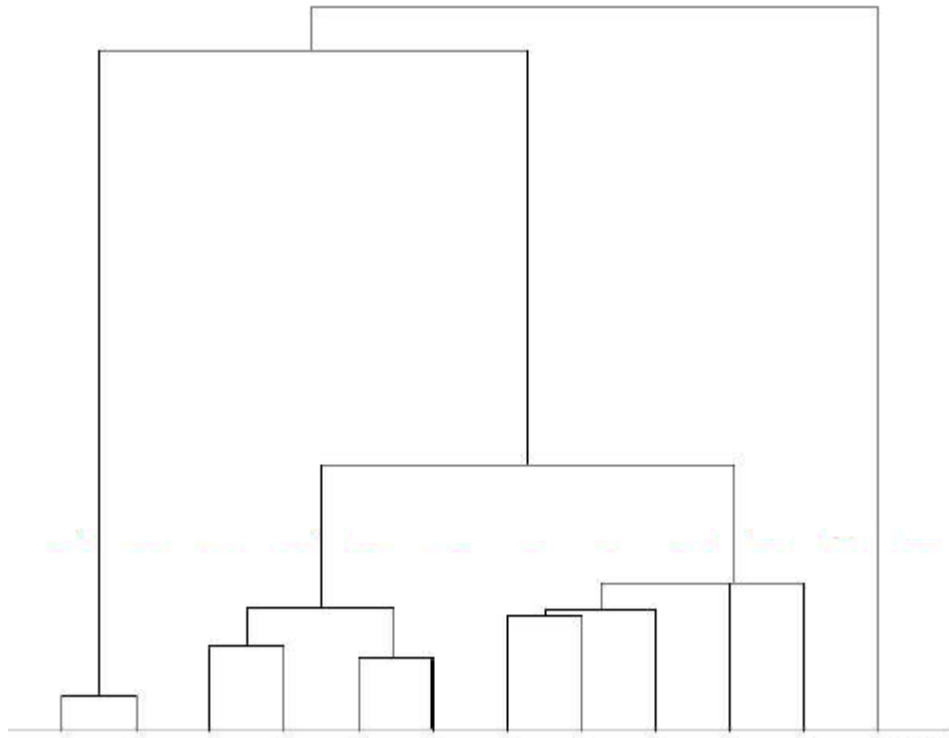


Fig.1.9 Schematizzazione del risultato di un clustering gerarchico.

Una volta ottenuto l'albero di fig.1.9 si sceglie un punto in cui “tagliare” l'albero; questa scelta determina il numero di clusters.

1.3.3.7 Clustering Incrementale

Partendo da un cluster contenente un elemento da clusterizzare, si aggiungono al gruppo gli elementi più vicini ad esso. Si smette di aggiungere elementi al cluster secondo un criterio che dipende dall'applicazione.

Nell'articolo [11] il metodo incrementale viene utilizzato per semplificare la rappresentazione di oggetti tridimensionali; in questo caso si smette di aggiungere elementi al cluster quando il numero degli elementi del cluster supera una certa soglia.

Nel capitolo 5 utilizzeremo il clustering incrementale, in quel caso smetteremo di aggiungere elementi al cluster quando non vi saranno più elementi vicini agli elementi già presenti nel cluster.

1.3.3.8 Metodi a potenziale

I metodi a potenziale consistono nello scegliere una funzione che assegna ad ogni elemento da clusterizzare un valore; tale valore rappresenta il potenziale che quell'elemento ha di diventare centroide del cluster. Gli elementi che hanno un potenziale alto diventano centroidi dei cluster. Non richiedono conoscenza a priori del numero di clusters.

1.3.3.9 Clustering per l'individuazione di oggetti

Per rendere conto dello stato della conoscenza per quanto riguarda il clustering di punti tridimensionali provenienti da un algoritmo di visione per l'individuazione di oggetti passiamo ad analizzare alcuni recenti articoli.

Nell'articolo [12] datato 2004, viene affrontato il problema della clusterizzazione di immagini di profondità. Le immagini di profondità sono immagini che contengono l'informazione della profondità di ogni pixel. Il calcolo di tali immagini può essere effettuato tramite visione, ad esempio utilizzando la funzione stereoscopica oppure utilizzando altre tecniche. L'applicazione per cui è stata progettata la tecnica presentata nell'articolo è il riconoscimento ostacoli in ambienti esterni: l'obiettivo sarà clusterizzare le immagini di profondità in modo da distinguere le zone dell'ambiente non navigabili da quelle navigabili.

Il sistema utilizza un algoritmo di clustering che prevede la conoscenza a priori del numero di cluster k ; si esegue una clusterizzazione per vari valori di k e si sceglie il k ottimo in base a criteri euristici (paragrafo 4 dell'articolo). Utilizzando il metodo appena descritto con un numero basso di punti da clusterizzare si ha il problema dell'overfitting. Tale problema non è menzionato nell'articolo perché le immagini di profondità hanno un numero di punti grande rispetto al valore di k che, nelle prove sperimentali presentate nell'articolo, ha un valore massimo pari a 10.

Nel caso che i punti tridimensionali in ingresso all'algoritmo siano un numero più basso di quello che hanno le immagini di profondità, il problema dell'overfitting si presenta necessariamente (par. 1.3.3.3).

Stando a quanto scritto nell'articolo, il funzionamento dell'algoritmo si basa sull'*assioma 1* (paragrafo 3.1 dell'articolo); ciò significa che l'insieme di punti tridimensionali in ingresso deve essere fatto in modo che "le strutture planari giochino un ruolo importante".

Per le considerazioni sopra espresse, l'algoritmo non può essere applicato nel caso in cui si abbia un numero di punti parecchio inferiore rispetto a quelli che solitamente si hanno in un'immagine di profondità.

Nell'articolo [13], datato 2004, si utilizza un sistema di visione monoculare. Ci si pone come obiettivo quello di identificare gli oggetti presenti nell'ambiente per distinguerli dallo sfondo dell'immagine. Si utilizza un algoritmo che estrae features dalle immagini. L'articolo si focalizza sulla selezione delle features che appartengono agli oggetti. Per stimare la posizione 3D delle features, ci si basa sulle posizioni bidimensionali che le features assumono sull'immagine nel tempo; il sistema, infatti, prevede il movimento della telecamera ed una tecnica che permette di stimare tale movimento. Per la stima della posizione 3D delle features si utilizza un filtro di Kalman esteso.

A questo punto, si presenta il problema di clusterizzare i punti tridimensionali per riconoscere gli oggetti senza sapere a priori il numero di cluster. Viene utilizzato il metodo gerarchico (par. 1.3.3.6). I risultati della tecnica implementata sono presentati in fig.1.10.

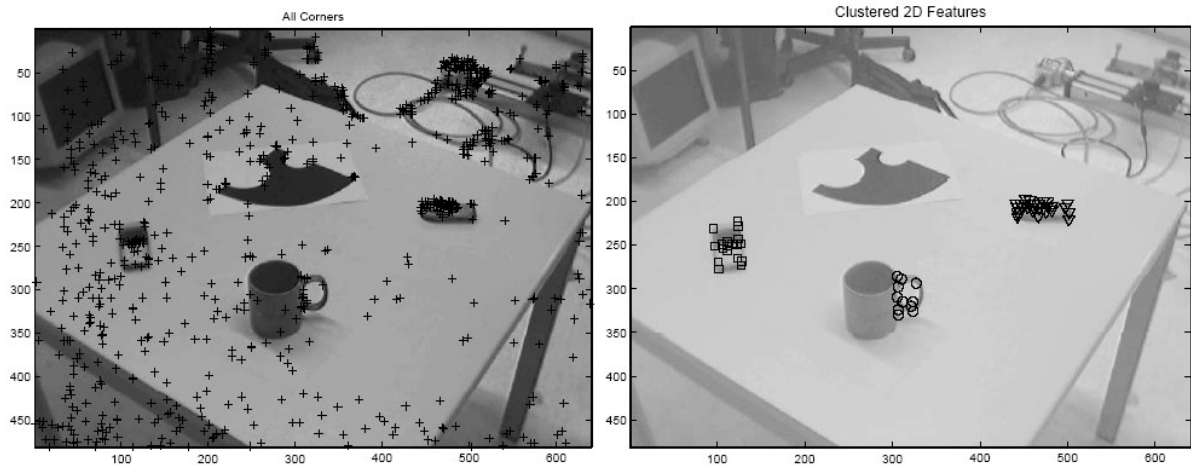


Fig.1.10 Algoritmo di clustering delle features nello spazio 3D

A sinistra: immagine di una sequenza video da cui sono state estratte features. Le features estratte sono contrassegnate con un '+'

A destra: le features sono state clusterizzate secondo la tecnica esposta e sono state visualizzati i cluster che contengono un certo numero di features.